Data Efficient Training for Materials Property Prediction Using Active Learning Querying

Carmelo Gonzales Intel Labs

Kin Long Kelvin Lee Intel DCAI

Bin Mu Intel TD

Mikhail Galkin Intel Labs

Santiago Miret Intel Labs

Abstract

The field of machine learning for materials property prediction and characterization is seeing rapid developments in models, datasets, and frameworks. While datasets and models grow in size, frameworks must mature concurrently to match the data requirements and quick development cycles required to support these growing workloads. The efficient training of models is one area where machine learning frameworks may be improved. Utilizing active learning querying strategies to train models from scratch using fewer data can lead to faster development cycles, model evaluations, and reduced costs of training. Well-studied active learning querying strategies from computer vision and natural language processing are directly applied to train an E(n)-GNN model from scratch using a subset of the Materials Project Database and Novel Materials Discovery (NOMAD) Database, with the results compared to data subset selection techniques and the standard training pipeline. In general, the models trained with active learning querying strategies meet or exceed the performance standard trained models while using significantly less training data.

1 Introduction

In the field of materials science, the ability to accurately predict material properties is a critical prerequisite for the design and synthesis of novel materials with tailored functionalities. Traditional computational methods relying on first-principles calculations or empirical relationships often face challenges in terms of computational cost, scalability, and generalizability. As the dimensionality and complexity of material datasets data grow, there's an increasing demand for innovative approaches to predict material properties efficiently and reliably Goodall and Lee [2020]Jha et al. [2019]. While machine learning models hold the potential to accelerate material property predictions Pilania et al. [2013], their effectiveness is inherently tied to the quality and comprehensiveness of the training data. Given the vastness of the materials space and the high cost associated with experimental and computational data generation, obtaining exhaustive labeled datasets becomes a significant bottleneck Jha et al. [2022]De Breuck et al. [2021][Song et al., 2023].

In typical machine learning workflows, models are trained by randomly sampling data from the full dataset until all samples are seen. Random sampling ensures the full dataset is seen in an unbiased way and ensures the full dataset distribution is well represented. Although beneficial, random sampling an entire dataset may be slow and redundant, as the dataset may contain samples that are not impactful on the model training and optimization itself. Long training times become a hindering factor in development as larger datasets are curated and models grown in both size and complexity. Reducing the dataset size requirements to achieve desirable model performance will allow for faster development cycles, increase accessibility to modern experimental pipelines, and give researchers more time to focus on other aspects of materials modeling challenges.

To mitigate these challenges, two types of data sampling techniques are explored to reduce overall dataset size requirements in training models from scratch. Data subset selection aims to find the

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

most informative samples such that a model trained with a subset may achieve similar performance to a model trained on the full dataset. Samples are ordered by their importance and iteratively fed to the model during training. The ordering is determined by strategies such as measuring distance of samples from their cluster centers in embedding space, uncertainty based scores, and feature similarityDasgupta and Hsu [2008]Lewis and Gale [1994].

Active learning is a related problem originally developed for semi-supervised learning scenarios where an existing model is to be updated with new data. In active learning querying, a pool of unlabeled data is analyzed for the most important subset of samples which are selected and subsequently labeled, typically by a human-in-the-loop or other oracle. Unlabeled samples are queried by strategies that are uncertainty or diversity based Settles [2011]. Newly queried samples are used to update the existing model by way of retraining or fine-tuning the model. In both data subset selection and active learning querying, building a subset of data that accurately captures the distribution and diversity of the original dataset is desired. The subset of data may then be used in place of the full dataset to perform more efficient training of models.

2 Background and Related Work

Active learning has been extensively applied to the computer vision and natural language processing domainsLi and Guo [2013]Wu et al. [2022]Zhang et al. [2022]. While active learning is typically applied in the semi-supervised setting, many prior works highlight the utility of using active learning query strategies to train models from scratch Chen et al. [2022]. The comprehensive work of Park et al. [2022] presents how computer vision models may be trained from scratch more efficiently using active learning querying compared to data subset selection methods, and highlights the importance of an initial random subset of data used as the initial training pool. Bengio et al. [2009] develop curriculum learning, where annotated training data is sorted from easy to learn, and is presented to the model during training.

While active learning has broad coverage in computer vision and natural language processing, the emerging field of applying deep learning in materials science has not been as thoroughly studied under the framework of active learning. Wang et al. [2020] demonstrates how active learning may be used in molecular dynamics simulations, where adversarial attacks move atoms towards uncertain positions based on forces, which are then used in active learning updates. Lookman et al. [2019] investigates the use of common querying functions and their use in materials science applications. Jain et al. [2023] looks at active learning from the perspective of multi-objective optimization. Graff and Coley [2022] develops a software library that accelerates the structure based virtual screening process for used in drug discovery programs. In the work of Siemenn et al. [2022], two new active learning acquisition functions are implemented which help guide sampling of new experiments. Bo et al. [2022] implements a multi-objective active learning algorithm for designing 3D-printed architected materials.

3 Querying Methods

Querying methods from active learning typically rely on the use of a model which selects data in an unsupervised manner, i.e., data labels are assumed to be unknown. In data subset selection, the full dataset and labels are available to see and utilize. Data subset selection may also make use of known features of a dataset which can be used to separate out useful samples. These querying methods are used in the context of pool based active learning, where multiple samples are selected at once, as opposed to stream based where samples are provided to a model one-by-one.

3.1 Data Subset Selection Methods

Three data subset selection methods are used, namely: random sampling, center of mass (COM) clustering and UMAP (Uniform Manifold Approximation and Projection) McInnes et al. [2018] clustering. In random sampling, data points are randomly selected from the pool of unqueried data. In center of mass clustering, the entire dataset is processed and local center of mass coordinates are generated for each sample. The average center of mass across the dataset is then used as the global center, and each sample is sorted by the Euclidean distance from the global center of mass. Similarly, for UMAP, the 2-D output global center is calculated, and individual samples are sorted by their distance from the center. During training, samples furthest out are selected first, and the closest points are selected last.

3.2 Active Learning Querying Methods

Standard active learning querying methods are used along with common variations including: entropy Holub et al. [2008], margin Roth and Small [2006], least confidence Lewis and Gale [1994], expected entropy Holub et al. [2008], expected least confidence Settles [2009], expected margin Settles [2009], and predictive entropy Hernández-Lobato et al. [2014]. For precise definitions of the standard strategies used please see Lesci [2022]. BALD Houlsby et al. [2011] is also utilized, which is the difference between predictive entropy and expected entropy. Entropy strategies used in this work are based on the entropy of the softmax of the logits. Confidence is defined as the maximum probability assigned to a class, as determined by the softmax. Finally, learning loss (LL) Yoo and Kweon [2019] is utilized which involves training a loss prediction module which shares the backbone of the primary model, and is trained to predict the losses of unlabeled inputs.

4 Experiments

4.1 Setup

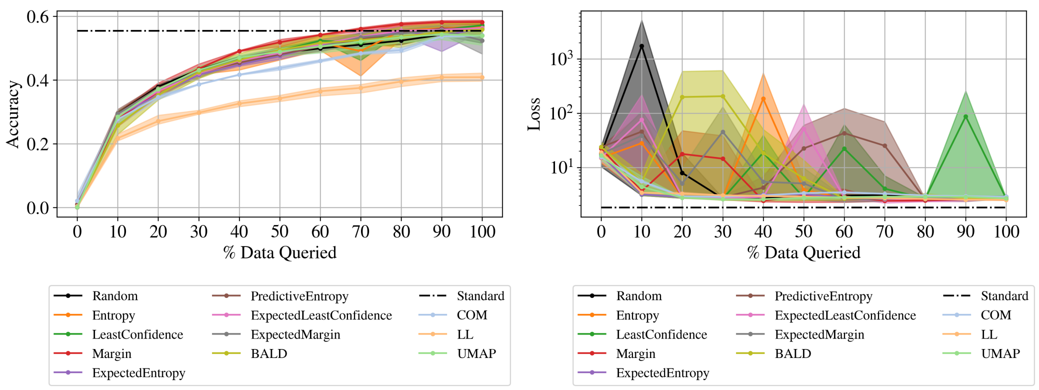
All experiments are performed utilizing the Open MatSciML Toolkit by Lee et al. [2023], and Energizer: an active learning framework for PyTorch based on PyTorch Lightning Lesci [2022]. Datasets used include the Materials Project Dataset [Jain et al., 2013] with train, validation and test splits defined in the Open MatSciML toolkit (108,159 train samples, 30,904 validation samples 15,456 test samples) and NOMAD Draxl and Scheffler [2019] randomly created train, validation, and test splits (97,174 train samples, 27,763 validation samples, 13,883 test samples). The E(n)-GNN model Satorras et al. [2021] implementation from Open MatSciML toolkit is used with default parameters for the model and optimizer. The crystal symmetry property consisting of 230 possible classes is used as the classification task to demonstrate the capabilities of active learning techniques.

Data is queried iteratively in chunks of 10% of the original training dataset size until the full dataset has been sampled. At each update, the full dataset queried up to the current update step is used to train the model. Each model update is trained for 25 epochs, and a total of 10 model updates are performed. At the end of each model update, the test set is used to generate model performance metrics. Three full experiments are run for each method using three random seeds per experiment. In the result plots, average classification accuracy and loss is plotted as a solid line. Experiments using active learning querying and data subset selection start with a randomly selected set of data which is shown to lead to better overall performance Park et al. [2022]. In each model querying iteration, the entire pool of available data is used and ranked based on the querying method. Selected data is subsequently added to the train dataloader and removed from the pool of available data to query from. All experiments are run on a cluster and utilize either a single Nvidia Titan-X, Titan-Xp or Titan-V GPU.

In additional to query strategies, model weight resetting is used, where the models weights are fully reset to their original initialization values after each active learning update. This type of experiment is closely related to training a model from scratch using smaller data subset sizes, with the caveat in active learning querying of using a previously trained model to aid in the selection of the next set of data. In the typical active learning setting, pretrained model weights are fine-tuned using small pools of oracle labeled data from active learning querying. This paradigm of active learning works well for incremental updates with queried datasets that are small in comparison to the original pretraining dataset Ren et al. [2021].

4.2 Results

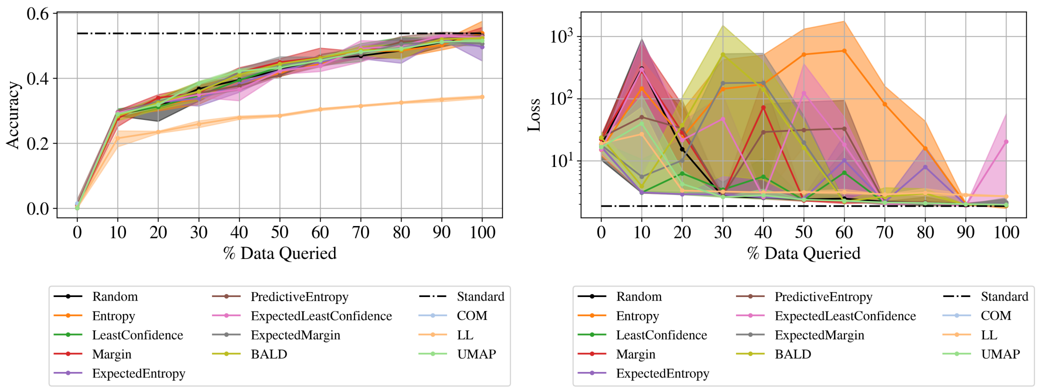
In general, active learning querying methods enable models to match the performance of standard training, and in some cases outperform the standard performance when comparing the test set classification accuracy. Additionally, active learning querying methods tend to outperform the data subset selection methods. The best results are seen when using a strategy that does not reset the model weights back to their original values. Figure 1 shows test accuracy and loss as the percentage of data sampled is increased.



(a) Test accuracy on Materials Project

(b) Test loss on Materials Project

Figure 1: Test set accuracy and loss curves for each query strategy, compared to standard training (dashed black line). These results did not reset the models weights between queries. All active learning strategies, except for learning loss, achieve equal or greater performance compared to standard training. Learning loss performance may be improved by tuning the loss prediction modules hyperparameters.



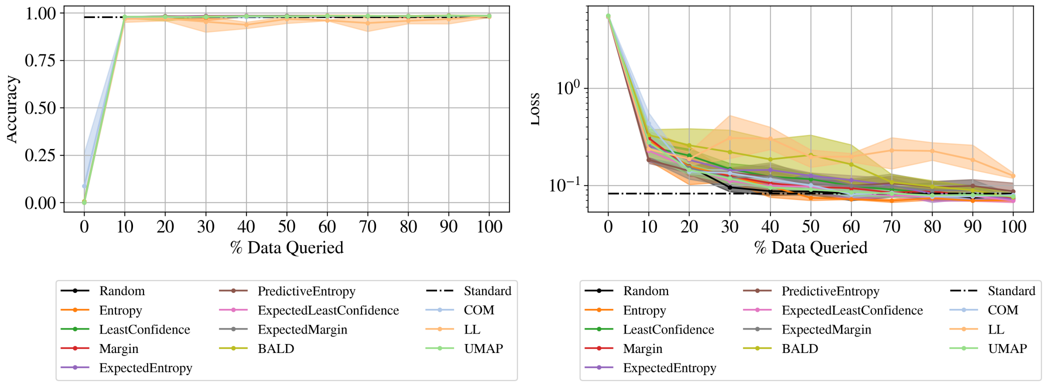
(a) Test accuracy on Materials Project

(b) Test loss on Materials Project

Figure 2: Test set accuracy and loss curves for each query strategy, compared to standard training (dashed black line). These results reset the model weights between queries. All strategies except learning loss converge to the same final performance as expected. Ideally, querying strategies would be able to select informative samples such that a model trained from scratch with a subset of data would achieve the same performance as a model trained on the full dataset.

Model performance remains consistent with standard training when the models weights are reset between queries. This is an indication that performing training simply with a smaller subset of data will not provide any performance gains compared to training with the full dataset. Figure 2 highlights how each strategy eventually converges to the standard model performance once the full dataset is sampled, except for learning loss, which never reaches the same final value. This may be due to a poor hyperparameter configuration for the learning loss module itself.

Experiments performed with NOMAD show results that indicate smaller datasets may be used to achieve the same performance as training with the full dataset, regardless of the method in which the data were sampled. In comparison to the Materials Project Dataset, the distribution of NOMAD space group numbers is far more sparse and can be seen in Appendix A Figure 5.



(a) Test accuracy on NOMAD

(b) Test loss on NOMAD

Figure 3: Test set accuracy and loss curves for each query strategy, compared to standard training (dashed black line). These results do not reset model weights. These results indicate that smaller datasets may be used to achieve the same performance as training with the full dataset, regardless of the method in which the data were sampled.

Table 1: Performance comparison of active learning querying strategies, data subset selection, and standard training using Materials Project dataset. Highlighted cells indicate a test set classification accuracy greater than or equal to the standard strategy.

Strategy	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Entropy	0.2 ± 0.3	28.6 ± 1.2	36.3 ± 0.5	42.4 ± 0.4	45.2 ± 1.5	48.7 ± 1.7	51.7 ± 1.7	49.1 ± 5.7	55.5 ± 2.0	56.3 ± 1.7	56.7 ± 2.1
Least Confidence	0.2 ± 0.1	29.4 ± 1.0	37.0 ± 0.6	43.0 ± 0.4	46.7 ± 0.2	49.3 ± 0.9	52.3 ± 0.3	51.7 ± 3.9	55.4 ± 0.4	55.8 ± 1.7	57.2 ± 0.3
Margin	0.1 ± 0.0	27.7 ± 1.9	36.4 ± 1.9	43.6 ± 1.4	49.1 ± 0.1	51.8 ± 1.0	54.2 ± 0.2	56.0 ± 0.5	57.5 ± 0.5	58.2 ± 0.7	58.2 ± 0.5
Expected Entropy	0.4 ± 0.4	28.7 ± 0.7	35.4 ± 0.5	41.6 ± 0.5	44.9 ± 0.3	47.6 ± 0.9	50.8 ± 1.4	52.8 ± 1.6	55.2 ± 0.8	53.9 ± 3.5	56.2 ± 0.7
Predictive Entropy	1.0 ± 1.2	29.5 ± 1.0	37.5 ± 0.8	42.4 ± 0.7	45.5 ± 0.7	48.0 ± 0.5	50.5 ± 1.1	52.8 ± 0.8	54.8 ± 1.5	56.2 ± 0.7	56.0 ± 0.9
Expected Least Confidence	0.4 ± 0.3	28.7 ± 0.3	36.5 ± 0.3	42.1 ± 0.6	46.0 ± 0.6	48.5 ± 0.8	51.2 ± 0.9	53.6 ± 1.2	55.5 ± 1.0	55.8 ± 1.2	56.3 ± 0.6
Expected Margin	0.2 ± 0.0	29.4 ± 0.2	37.2 ± 0.8	42.5 ± 1.5	46.9 ± 0.4	49.4 ± 0.6	50.5 ± 1.0	53.4 ± 1.2	54.6 ± 1.0	55.6 ± 1.2	52.4 ± 3.0
BALD	0.4 ± 0.2	25.8 ± 2.2	35.3 ± 1.5	42.5 ± 1.5	46.3 ± 1.4	49.4 ± 1.7	51.6 ± 1.7	52.6 ± 1.2	53.6 ± 2.7	55.1 ± 1.9	55.8 ± 1.5
Learning Loss	0.4 ± 0.3	21.8 ± 0.9	27.1 ± 1.3	29.8 ± 0.5	32.6 ± 0.8	34.3 ± 0.8	36.5 ± 1.0	37.5 ± 1.1	39.5 ± 1.1	40.9 ± 1.0	40.9 ± 1.0
Random	0.4 ± 0.2	29.5 ± 0.4	37.9 ± 0.5	43.3 ± 0.3	47.0 ± 0.2	48.4 ± 0.7	49.9 ± 0.3	51.1 ± 0.5	52.4 ± 1.0	54.2 ± 0.2	54.0 ± 1.7
Center of Mass	2.1 ± 2.1	27.7 ± 1.8	34.7 ± 0.7	38.7 ± 0.1	41.7 ± 0.2	43.7 ± 0.6	46.0 ± 0.4	48.4 ± 0.3	49.4 ± 0.8	53.7 ± 0.6	54.0 ± 0.1
UMAP	0.1 ± 0.0	28.5 ± 1.9	37.2 ± 0.3	43.1 ± 0.8	47.0 ± 0.8	48.9 ± 1.0	50.5 ± 1.4	52.0 ± 1.5	53.4 ± 1.1	54.3 ± 1.1	54.0 ± 2.1
Standard	-	-	-	-	-	-	-	-	-	-	54.0 ± 1.1

5 Discussion

We show that by using active learning querying, model classification accuracy may match or outperform standard performance while being trained with fewer data. All the active learning querying strategies, except for learning loss, achieve the same classification accuracy prior to that of a model trained with the full dataset and in some cases can achieve the same accuracy while using 40% less data. While the active learning query strategies perform well, data subset selection methods do not bring model performance up as quickly. When training with the NOMAD dataset, the same model performance was achieved by training with only 10% of the original dataset. Utilizing more efficient training methods in practice may allow researchers and practitioners to focus their efforts on model development, analysis, and dataset exploration.

While this work focuses primarily around active learning querying methods, a similar study may be conducted with a focus on data subset sampling methods to more thoroughly cover this space. Additionally, studying methods which train models for a comparable total number of epochs would lead to reductions in both data and compute requirements.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Peng Bo, Ye Wei, Yu Qin, Jiabao Dai, Liuliu Han, Yue Li, and Peng Wen. A data-efficient multiobjective machine learning method for 3d-printed architected materials design. In *AI for Accelerated Materials Design NeurIPS 2022 Workshop*, 2022. URL `https://openreview.net/forum?id=8B9urUw57L-`.

Liangyu Chen, Yutong Bai, Siyu Huang, Yongyi Lu, Bihan Wen, Alan L Yuille, and Zongwei Zhou. Making your first choice: To address cold start problem in vision active learning. *arXiv preprint arXiv:2210.02442*, 2022.

Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215, 2008.

Pierre-Paul De Breuck, Geoffroy Hautier, and Gian-Marco Rignanese. Materials property prediction for limited datasets enabled by feature selection and joint learning with modnet. *npj Computational Materials*, 7(1):83, 2021.

Claudia Draxl and Matthias Scheffler. The nomad laboratory: from data sharing to artificial intelligence. *Journal of Physics: Materials*, 2(3):036001, 2019.

Rhys EA Goodall and Alpha A Lee. Predicting materials properties without crystal structure: Deep representation learning from stoichiometry. *Nature communications*, 11(1):6280, 2020.

David E Graff and Connor W Coley. Molpal: Software for sample efficient high-throughput virtual screening. In *AI for Accelerated Materials Design NeurIPS 2022 Workshop*, 2022.

José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27, 2014.

Alex Holub, Pietro Perona, and Michael C Burl. Entropy-based active learning for object recognition. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1):011002, 2013.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-García, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective GFlowNets. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 14631–14653. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/jain23a.html`.

Dipendra Jha, Kamal Choudhary, Francesca Tavazza, Wei-keng Liao, Alok Choudhary, Carelyn Campbell, and Ankit Agrawal. Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning. *Nature communications*, 10 (1):5316, 2019.

Dipendra Jha, Vishu Gupta, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Moving closer to experimental level materials property prediction using ai. *Scientific reports*, 12(1):11953, 2022.

Kin Long Kelvin Lee, Carmelo Gonzales, Marcel Nassar, Matthew Spellings, Mikhail Galkin, and Santiago Miret. Matsciml: A broad, multi-task benchmark for solid-state materials modeling. *arXiv preprint arXiv:2309.05934*, 2023.

Pietro Lesci. Energizer: an active-learning framework for pytorch based on pytorch-lightning. https://github.com/pietrolesci/energizer/tree/v0.2.0, 2022.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. *CoRR*, abs/cmp-lg/9407020, 1994. URL http://arxiv.org/abs/cmp-lg/9407020.

Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 859–866, 2013.

Turab Lookman, Prasanna V Balachandran, Dezhen Xue, and Ruihao Yuan. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials*, 5(1):21, 2019.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Dongmin Park, Dimitris Papailiopoulos, and Kangwook Lee. Active learning is a strong baseline for data subset selection. In *Has it Trained Yet? NeurIPS 2022 Workshop*, 2022.

Ghanshyam Pilania, Chenchen Wang, Xun Jiang, Sanguthevar Rajasekaran, and Ramamurthy Ramprasad. Accelerating materials property predictions using machine learning. *Scientific reports*, 3(1):2810, 2013.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pages 413–424. Springer, 2006.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

Burr Settles. Active learning literature survey. 2009.

Burr Settles. From theories to queries: Active learning in practice. In Isabelle Guyon, Gavin Cawley, Gideon Dror, Vincent Lemaire, and Alexander Statnikov, editors, *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, pages 1–18, Sardinia, Italy, 16 May 2011. PMLR. URL https://proceedings.mlr.press/v16/settles11a.html.

Alexander E. Siemenn, Zekun Ren, Qianxiao Li, and Tonio Buonassisi. Accelerating the discovery of rare materials with bounded optimization techniques. In *AI for Accelerated Materials Design NeurIPS 2022 Workshop*, 2022. URL https://openreview.net/forum?id=9JK83z2mck.

Yu Song, Santiago Miret, and Bang Liu. Matsci-nlp: Evaluating scientific language models on materials science language tasks using text-to-schema modeling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

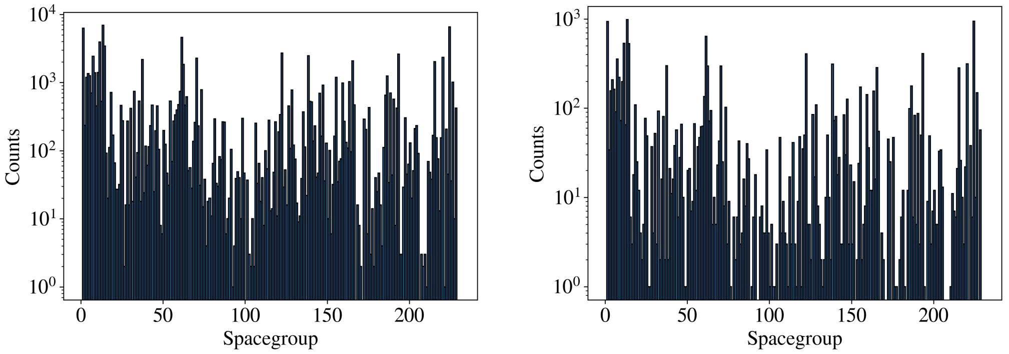
Wujie Wang, Tzuhsiung Yang, William H Harris, and Rafael Gómez-Bombarelli. Active learning and neural network potentials accelerate molecular screening of ether-based solvate ionic liquids. *Chemical Communications*, 56(63):8920–8923, 2020.

Mingfei Wu, Chen Li, and Zehuan Yao. Deep active learning for computer vision tasks: Methodologies, applications, and challenges. *Applied Sciences*, 12(16):8103, 2022.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. A survey of active learning for natural language processing. *arXiv preprint arXiv:2210.10109*, 2022.

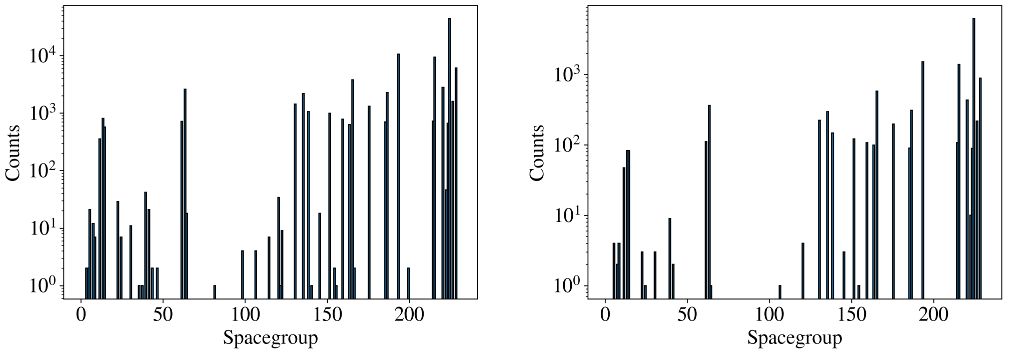
A Datasets and Distributions



(a) Train data distribution

(b) Test data distribution

Figure 4: Train and test set distribution over the space group property for the Materials Project dataset.



(a) Train data distribution

(b) Test data distribution

Figure 5: Train and test set distribution over the space group property for the NOMAD dataset. The distribution is far more sparse than Materials Project, despite having a similar number of total samples.

A.1 Hyperparameters

We outline the hyperparameters for E(n)-GNN described in Section 4. We maintained consistent architecture parameters for all training settings across all experiments.

Table 2: Hyperparameters for E(n)-GNN

Hyperparameter	Value
MLP hidden dim	32
MLP output dim	128
# of EGNN layers	5
Node MLP dim	$[128, 128, 128]$
Edge MLP dim	$[128, 128, 128]$
Atom position MLP dim	$[64, 64]$
MLP activation	ReLU
Graph read out	Sum
Node projection block depth	3
Node projection hidden dim	128
Node projection activation	ReLU
Output block depth	3
Output hidden dim	128
Output activation	ReLU
Optimizer Parameters	
Learning Rate	0.0001
Batch Size	24